Building your own USB camera with Gstreamer

Gstreamer Conference `23 – A Coruña

Michael Grzeschik – m.grzeschik@pengutronix.de

Who am I? – Michael Grzeschik

- @Pengutronix since 2009
- Embedded Linux Engineer

- Kernel Development
- Graphic/Media

Building your own USB camera

- Motivation
- Problem to be solved
- Preparations
- Solution → Spoiler: uvcsink!
- Open Topics and Next Steps

Motivation

- USB Video Class (UVC) Support for the Visualizer-Project

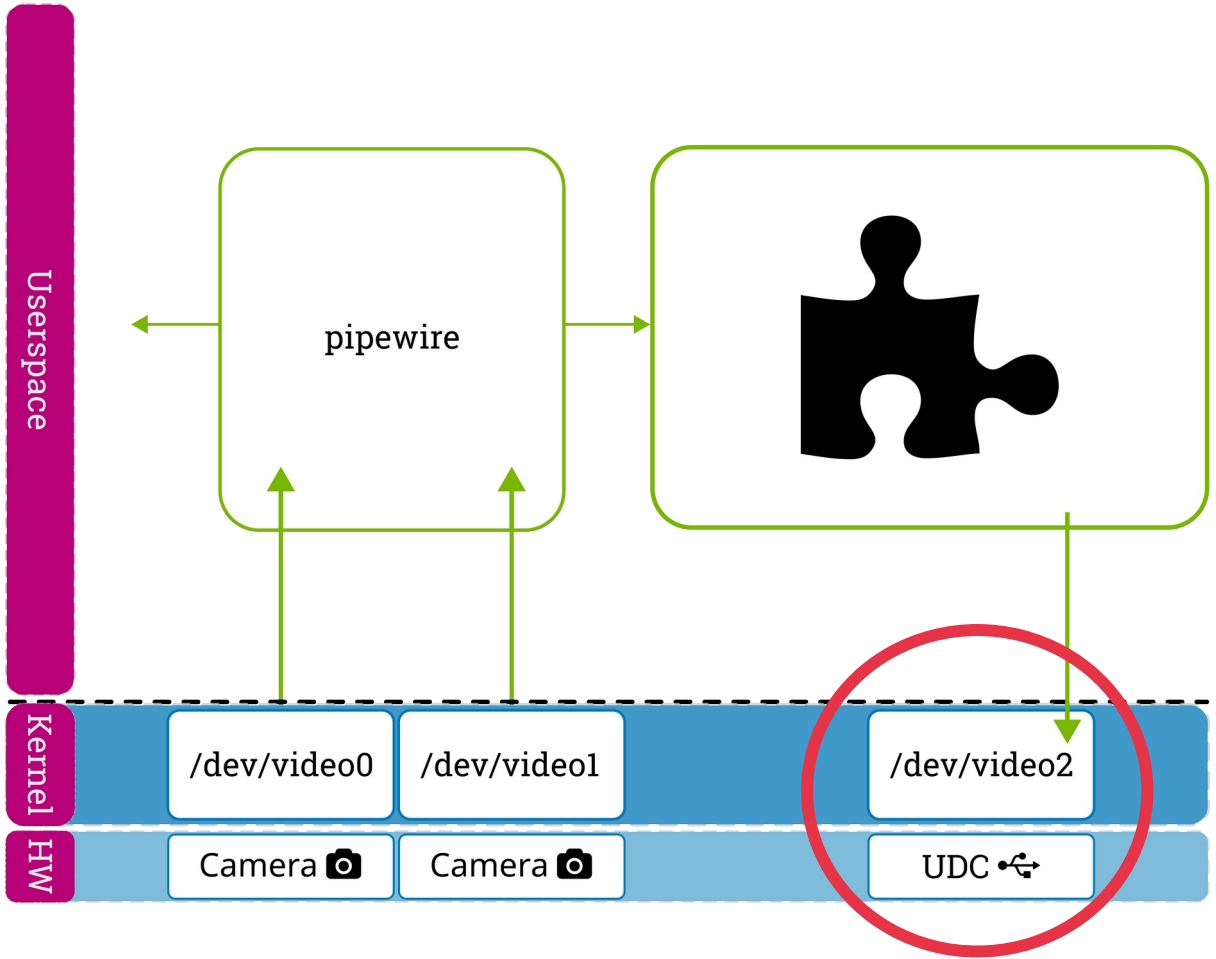
WOLFVISION®

Problem to be solved

- MJPEG USB Video Streaming
  - 1080p@30fps and 4k@30fps

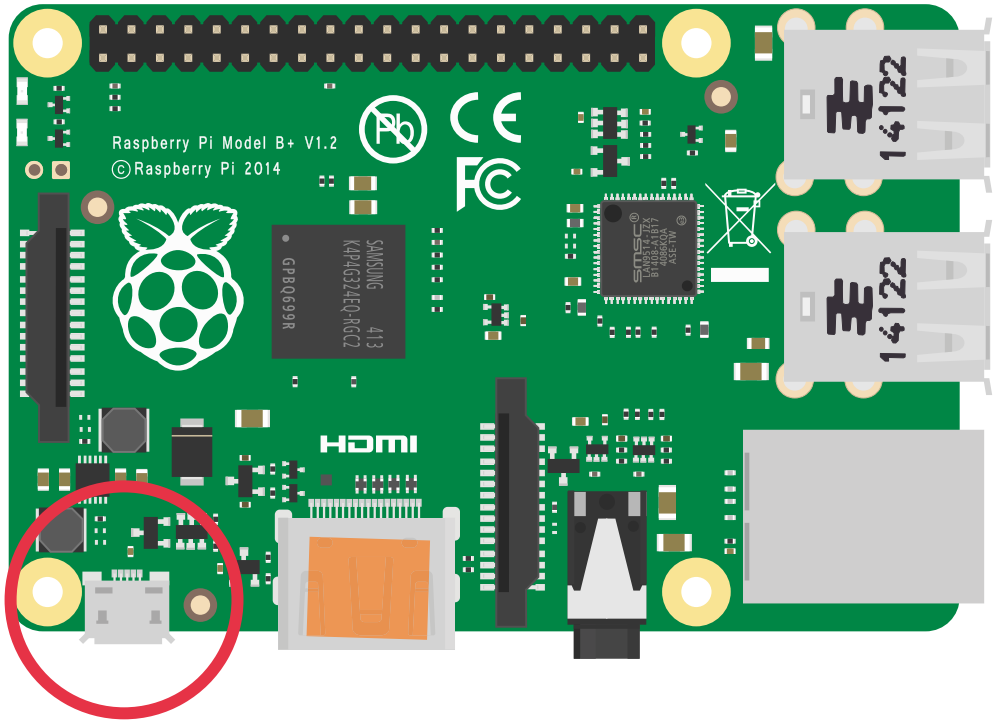
- both formats should be selectable at runtime
- simultanious stream the video over the network
- „just work like any other webcam“

Softwarearchitecture on the Project



Simplification of the Problem

- Raspberry-Pi USB-C Port → USB Gadget (2.0)



Preparation of the uvc gadget

Config-Filesystem (configfs) – setup

- setup attribute files in mounted directory
  - mount -t configfs none /sys/kernel/config/
  - mkdir /sys/kernel/config/usb_gadget/gadget/function/uvc.cam0
  - echo "...." > /sys/kernel/config/usb_gadget/gadget/function/...

Config-Filesystem (configfs) – binding

- binding the gadget with the USB Device Controler (UDC)
  - ls /sys/class/udc > /sys/kernel/config/usb_gadget/g1/UDC

Config-Filesystem (configfs) – scripting

- https://git.ideasonboard.org/uvc-gadget.git/
  - scripts/uvc-gadet.sh

- overall experience
  - no visual seperation of attribute values and executing script
  - hard to maintain, error prone

Config-Filesystem (configfs) – libusbgx

- libusbgx - https://github.com/linux-usb-gadgets/libusbgx
  - attribute structs represent key value pairs in configfs
  - export/import functions
  - many gadgets supported (mass-storage, serial, uac2, ...)
  - support for uvc was missing
    - added with https://github.com/linux-usb-gadgets/libusbgx/pull/64

Config-Filesystem (configfs) – gt

- gadget-tool (gt) - https://github.com/linux-usb-gadgets/gt
  - using libusbgx
  - scheme files for complete gadget description
    - easy to maintain
  - flexible for composite (multifunction) gadgets
    - e.g. combinations of uac2 and uvc, or ethernet with serial

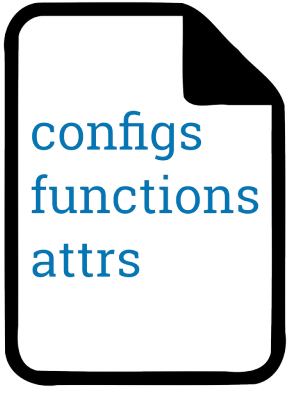
Config-Filesystem (configfs) – uvc.scheme

```
configs = (
{
  id = 1;
  name = "c";
  functions = (
  {
    name = "ecm.net";
    function = "ecm";
  }, {
    name = "uvc.cam0";
    function = "uvc";
  } );
} );
```

```
functions :
{
 ecm:
  {
    instance = "net";
    type = "ecm";
    attrs : { ... };
  };
 uvc:
  {
    instance = "cam0";
    type = "uvc";
    attrs : {};
};
```

```
attrs :
{
  idVendor = 0xABCD;
  idProduct = 0x1234;
};
strings = ( );
```

```
formats :
{
  mjpeg :
  {
    frames = (
    {
      dwFrameInterval = 333333;
      wHeight = 1080;
      wWidth = 1920;
    }, {
      dwFrameInterval = 333333;
      wHeight = 2160;
      wWidth = 3840;
    } );
  };
};
```



gadget-tool – setup and bind

- gt [-o] uvc.scheme [udc]
  - „-o“ : do not bind after load
- gt enable uvc.scheme [udc]

Config-Filesystem (configfs) – uvc

- if uvc gadget is included – v4l2 /dev/videoX is created
  - gadget not started until
    - software opened the device file
    - software subscribed to UVC host events

Serving the Video-Stream to the Gadget

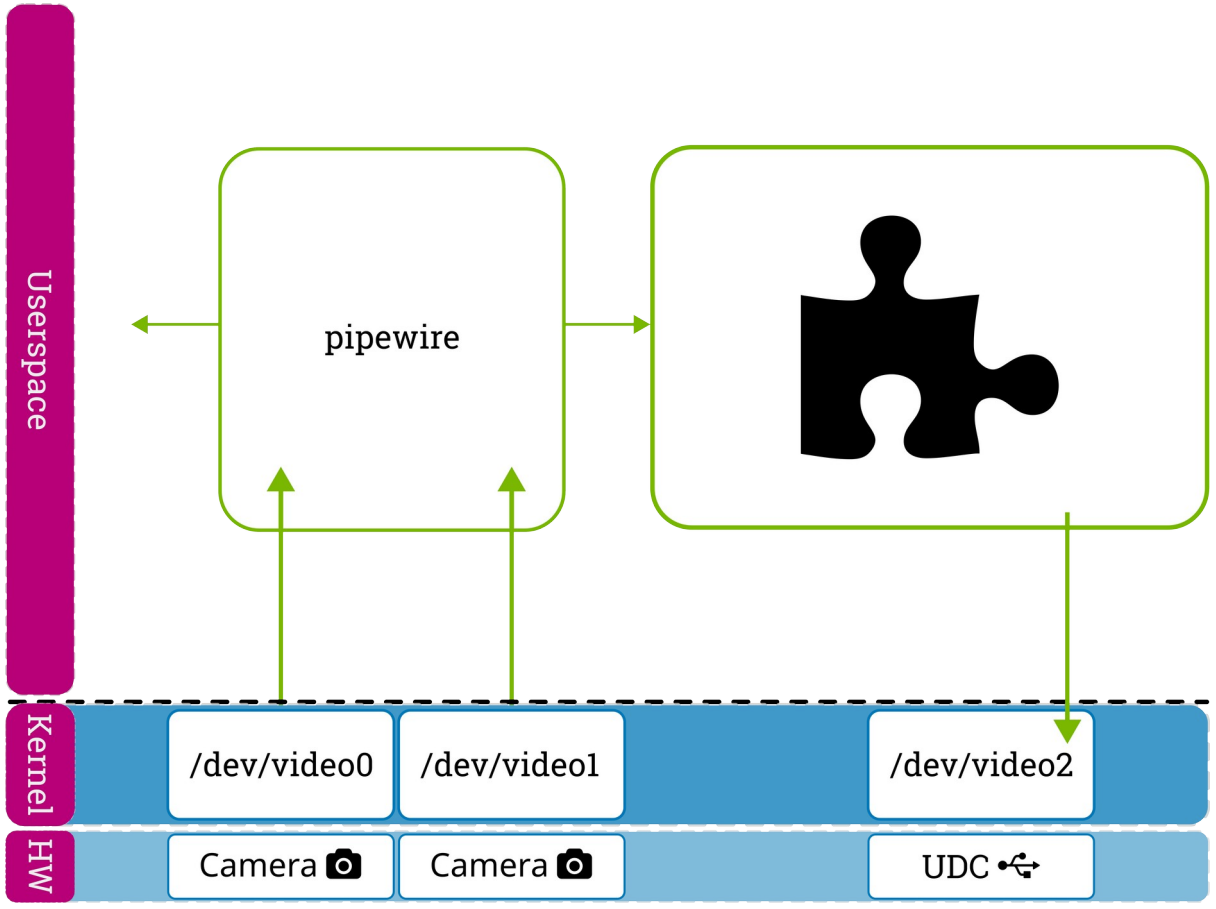
Streaming video to the host – application

- receive video stream from source
- serve video stream to gadget
- react to uvc host events
  - start/stop/format change

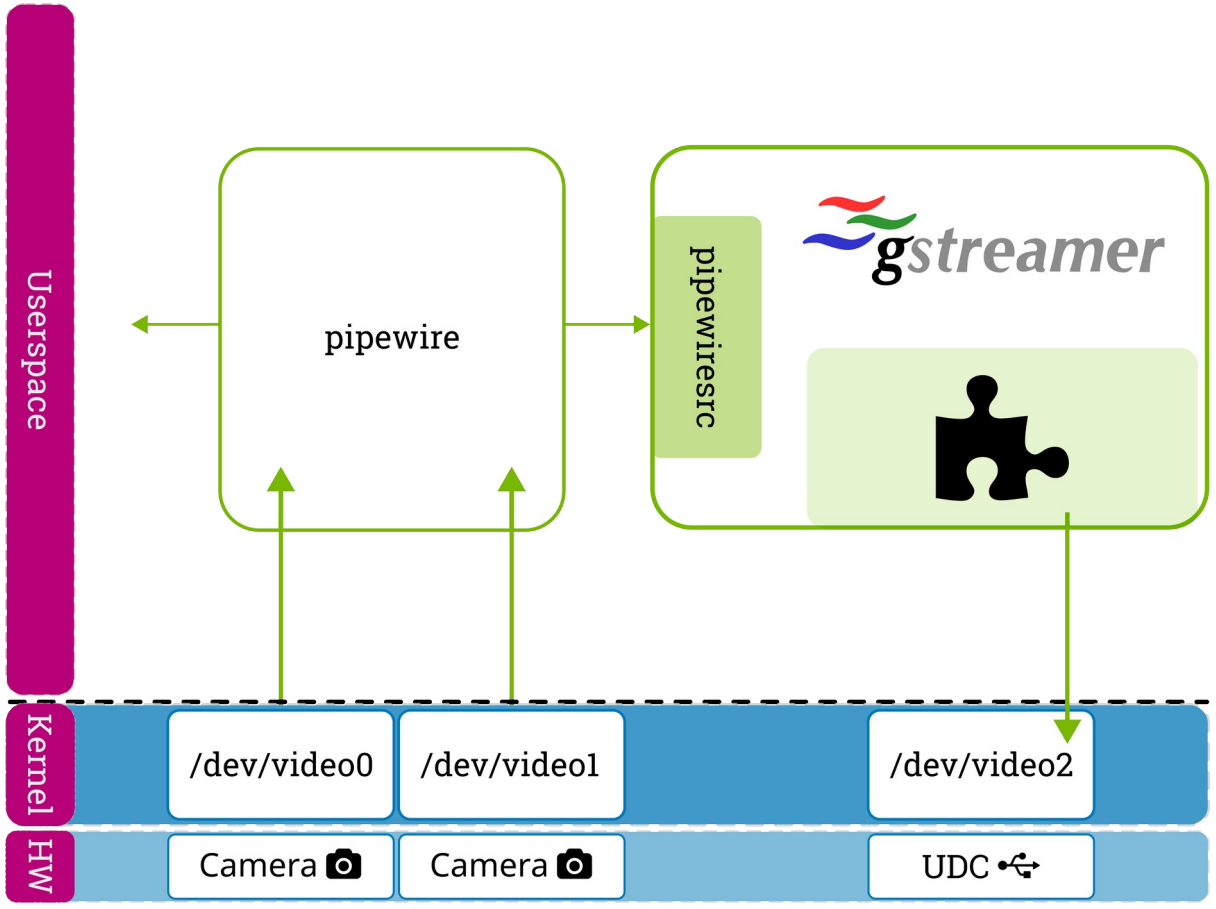
Streaming video to the host – uvc-gadget

- uvc-gadget - https://git.ideasonboard.org/uvc-gadget.git
  - uvc-gadget -c /dev/video2
  - many uvc host events are already handled
    - STREAMON, STREAMOFF, GET_CUR, SET_CUR, ...
  - dmabuf support for zero copy
  - possible to compile as library
  - different sources supported
    - v4l2
    - libcamera (lately)
  - gstreamer, pipewire → unsupported

uvc-gadget – does it match the project?

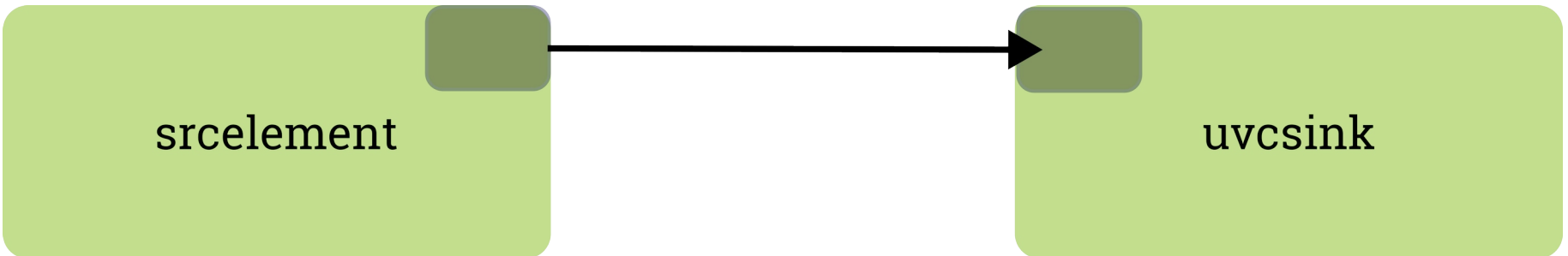


gstreamer – would it match the project?



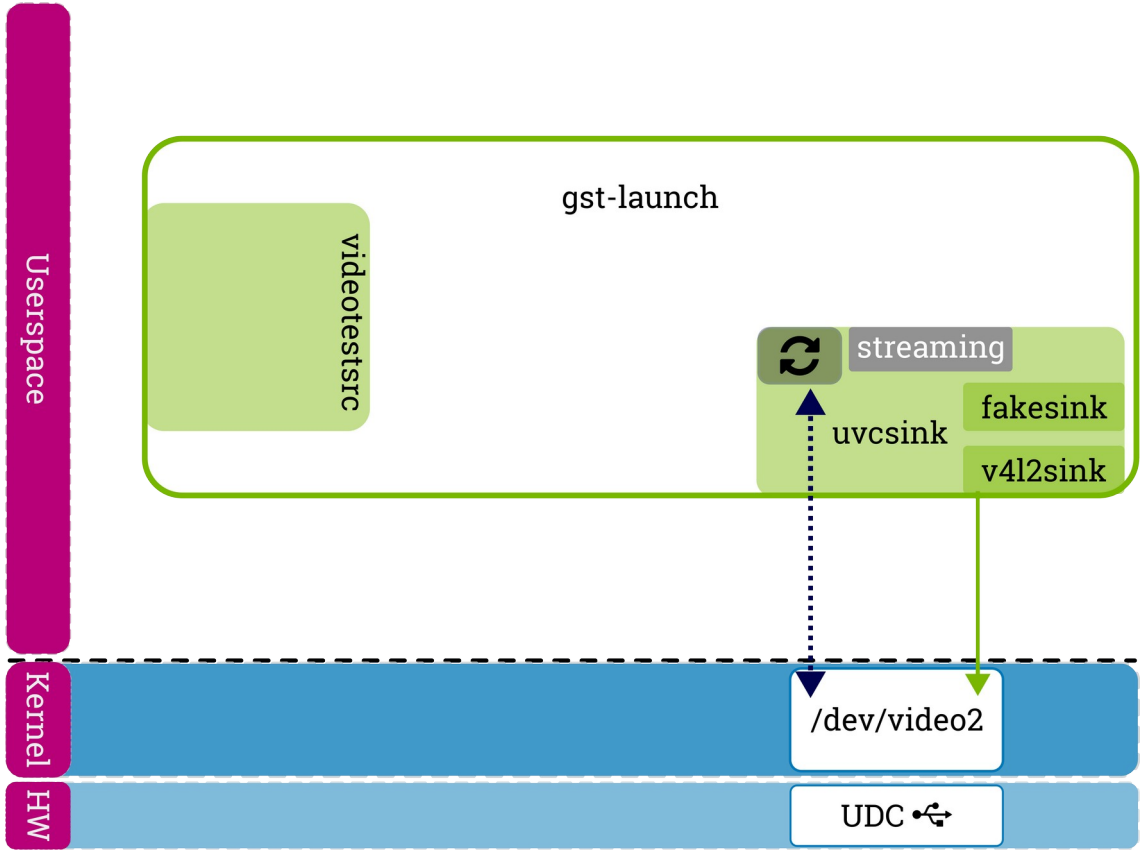
Gstreamer uvcsink element

Trivial setup – uvcsink



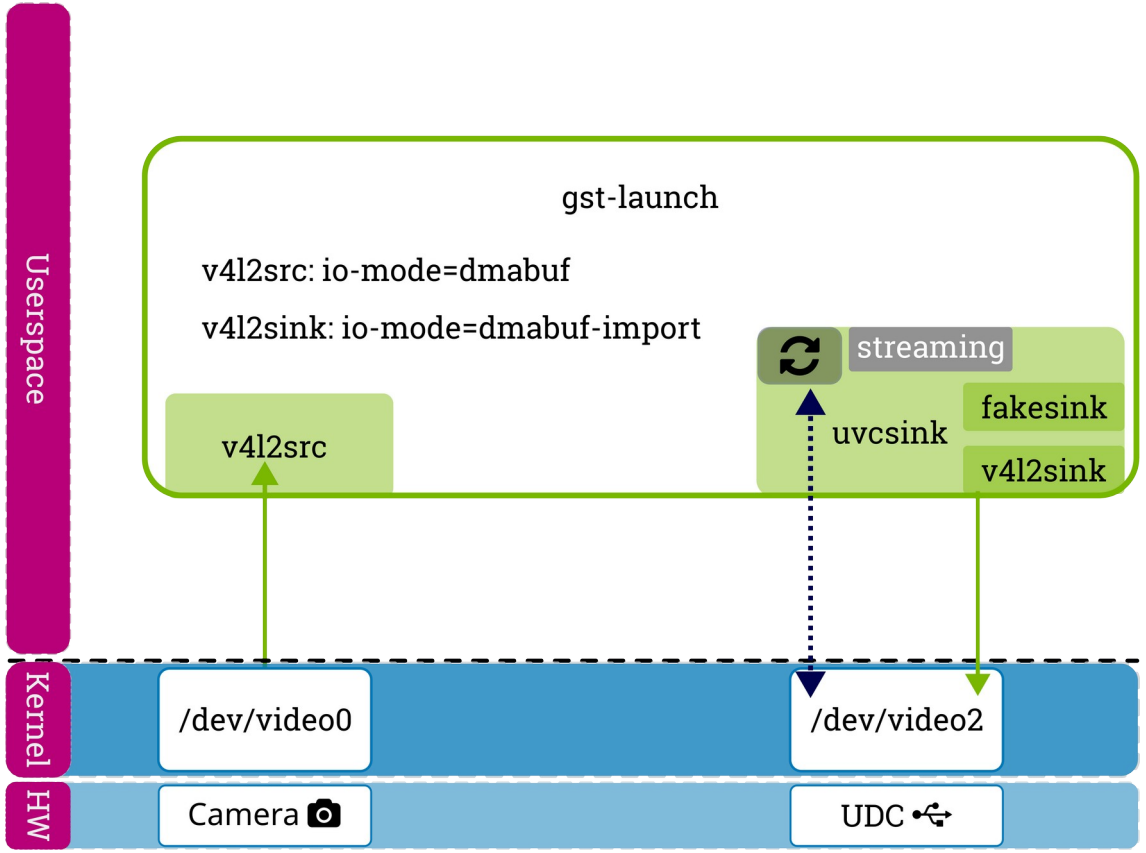
Trivial setup – uvcsink

gst-launch videotestsrc ! uvcsink v4l2sink::device=/dev/videoX



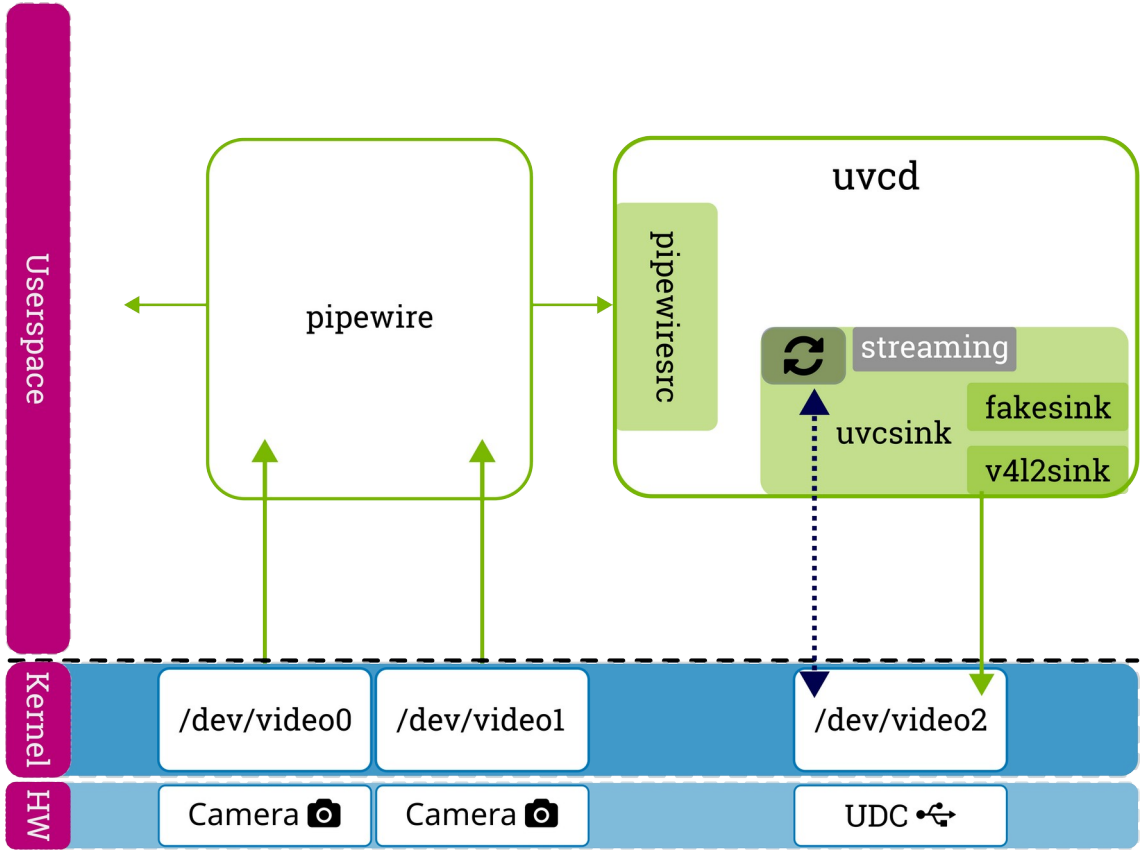
Trivial setup – uvcsink

gst-launch v4l2src device=/dev/videoY ! uvcsink v4l2sink::device=/dev/videoX



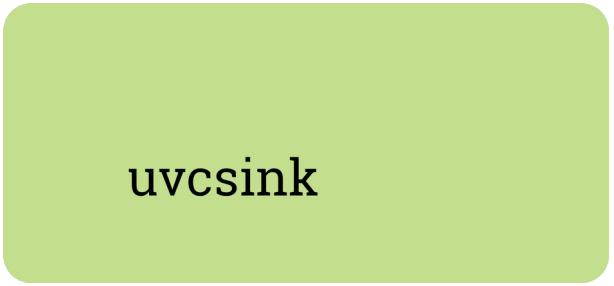
Customer setup – uvcsink

- pipewiresrc with uvcsink in customer application uvcd



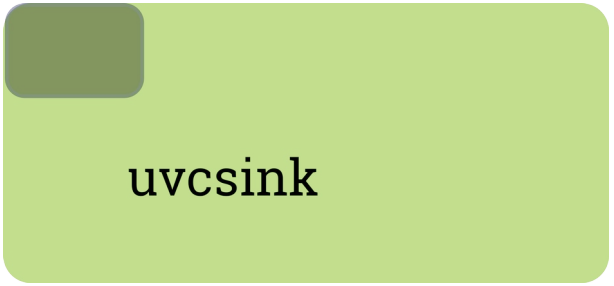
uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)



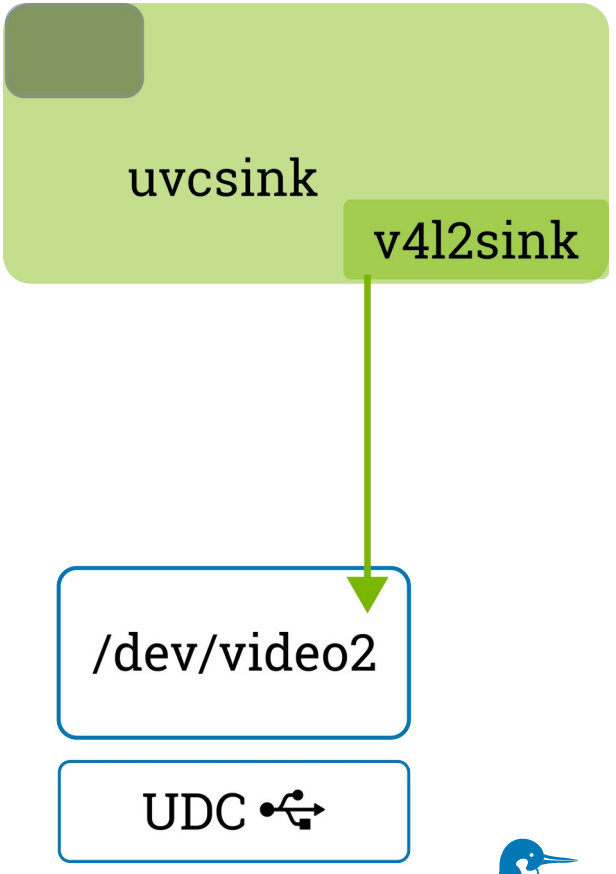
uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)



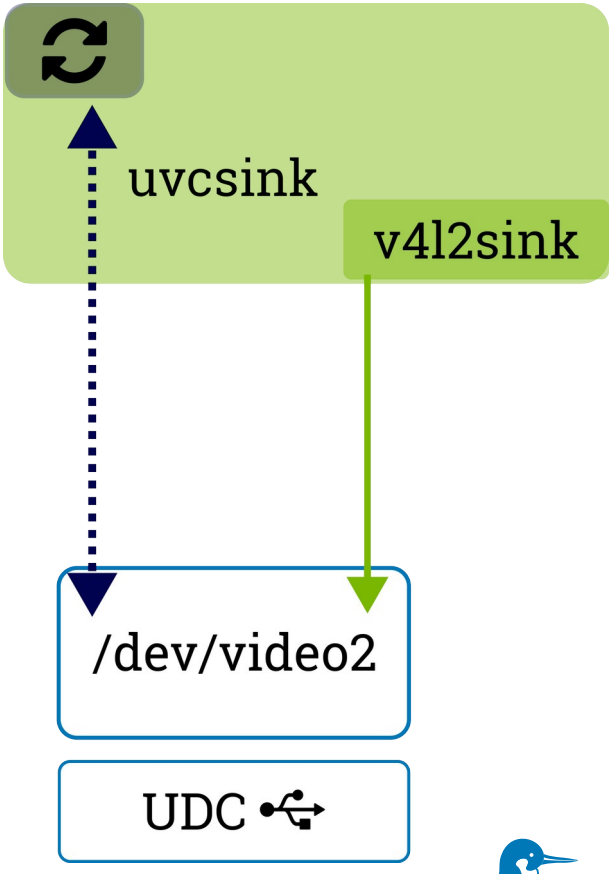
uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)



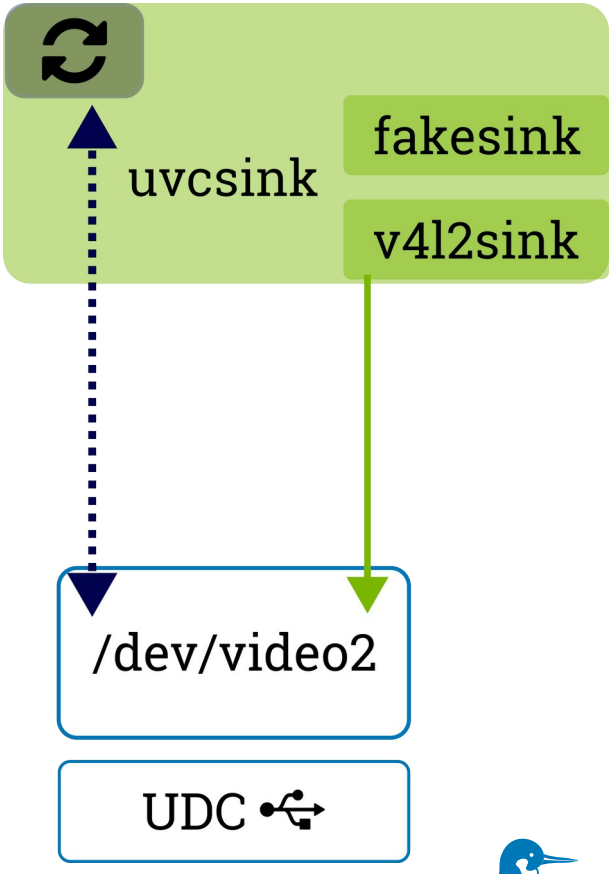
uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)



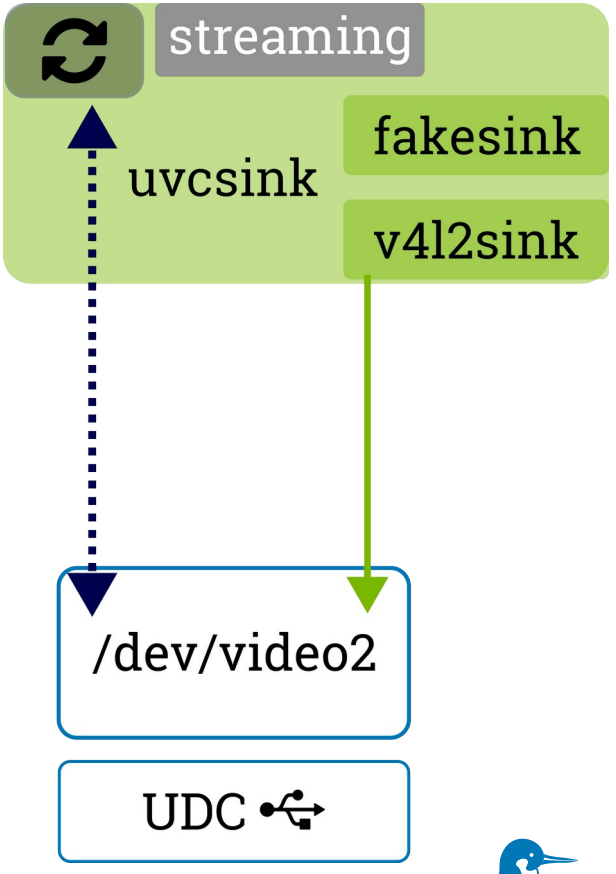
uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)



uvcsink – gst-plugins-bad/sys/uvcgadget

- bin
- one ghost sinkpad
- uses the v4l2sink
- host events in padtask
- uses the fakesink (sync)
- notifying property (streaming)

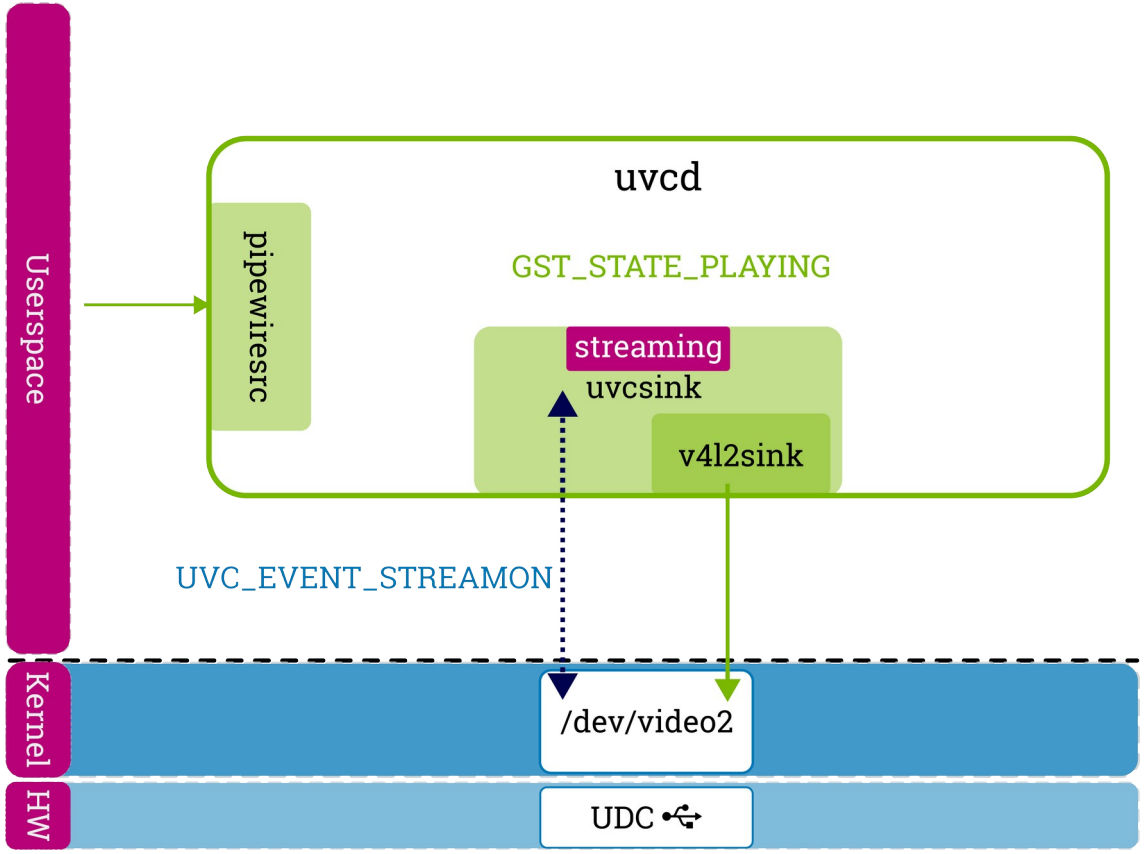


uvcsink – caps parsing in the v4l2sink

- Problem: v4l2 interface from gadget was not v4l2 compliant
  - probed_caps from v4l2sink was empty
- Solution: mainline missing v4l2 callbacks in the kernel
  - v4l2_enum_{format, framesizes, frameintervals}

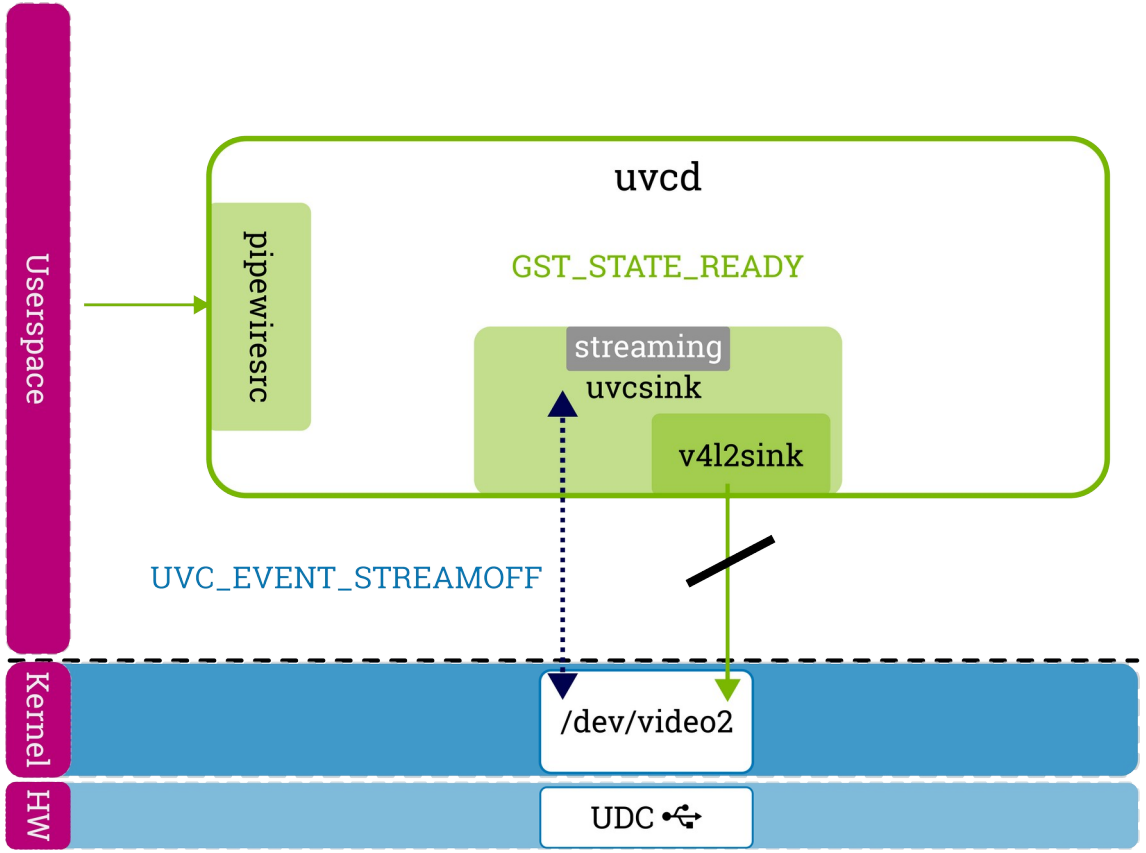
uvcsink – statefull mode

- implement property notification handling



uvcsink – statefull mode

- property streaming changed? – also change the GST_STATE

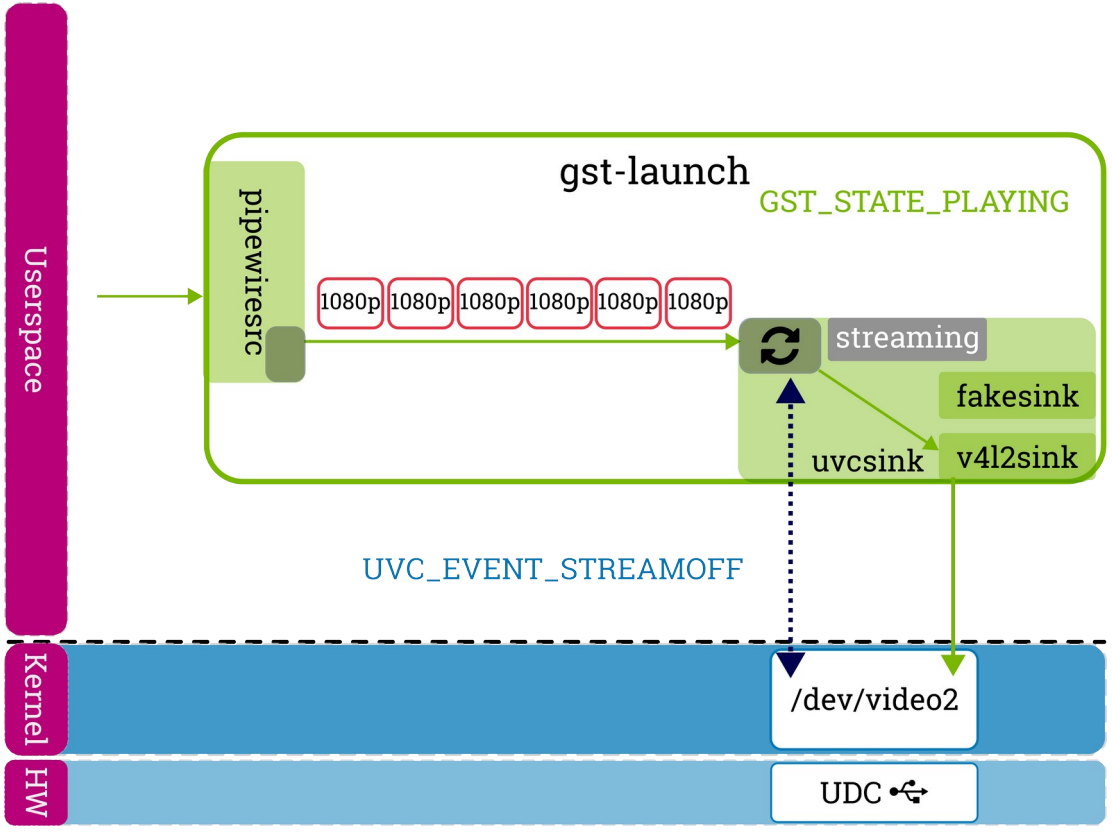


uvcsink – live mode (e.g. gst-launch)

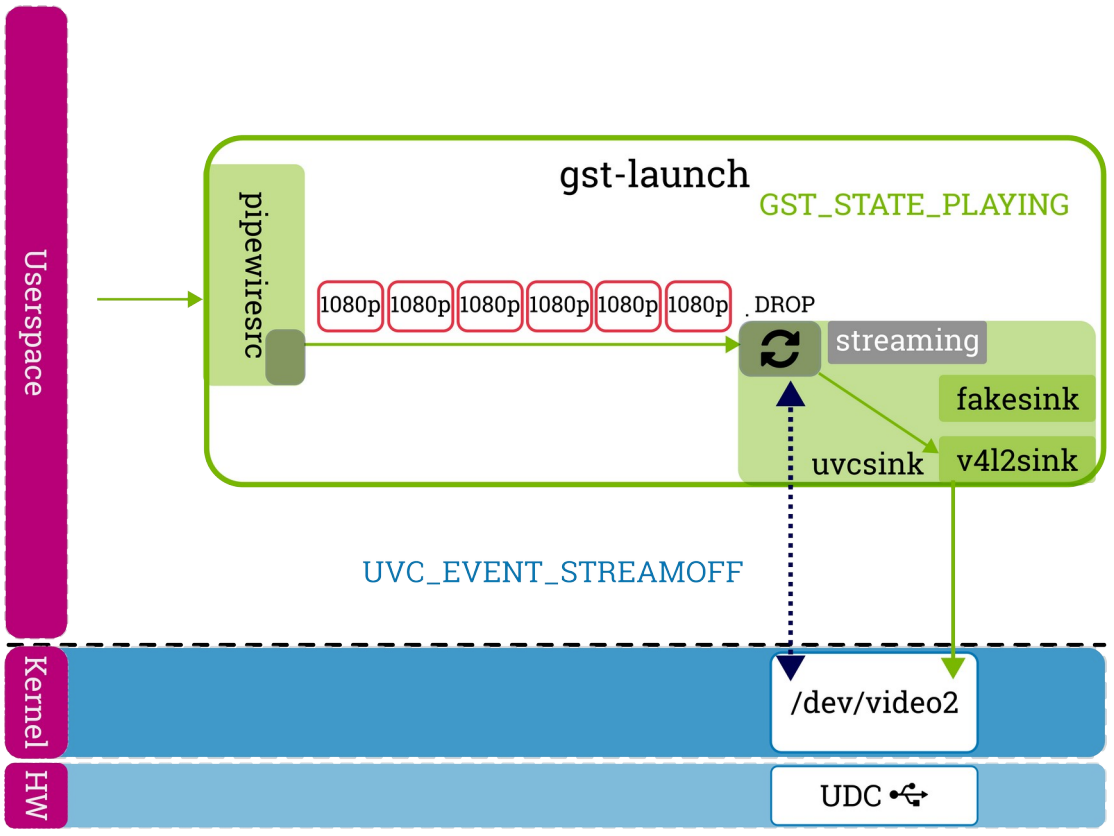
- pipeline is always in GST_STATE_PLAYING
- streaming event property is ignored

- How does the stream routing work?
  - bypass incoming frames to fakesink while host is not streaming
  - drop old incoming frames after an format change
  - switch to v4l2sink on demand

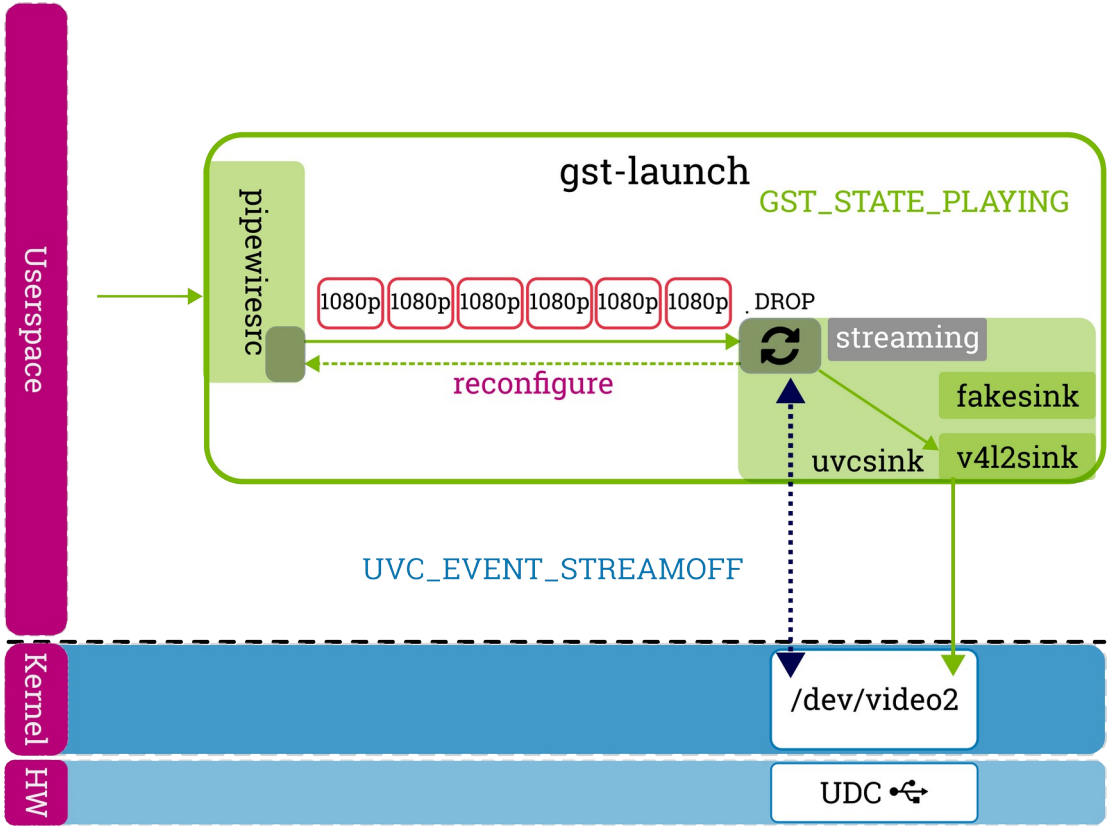
uvcsink – live mode



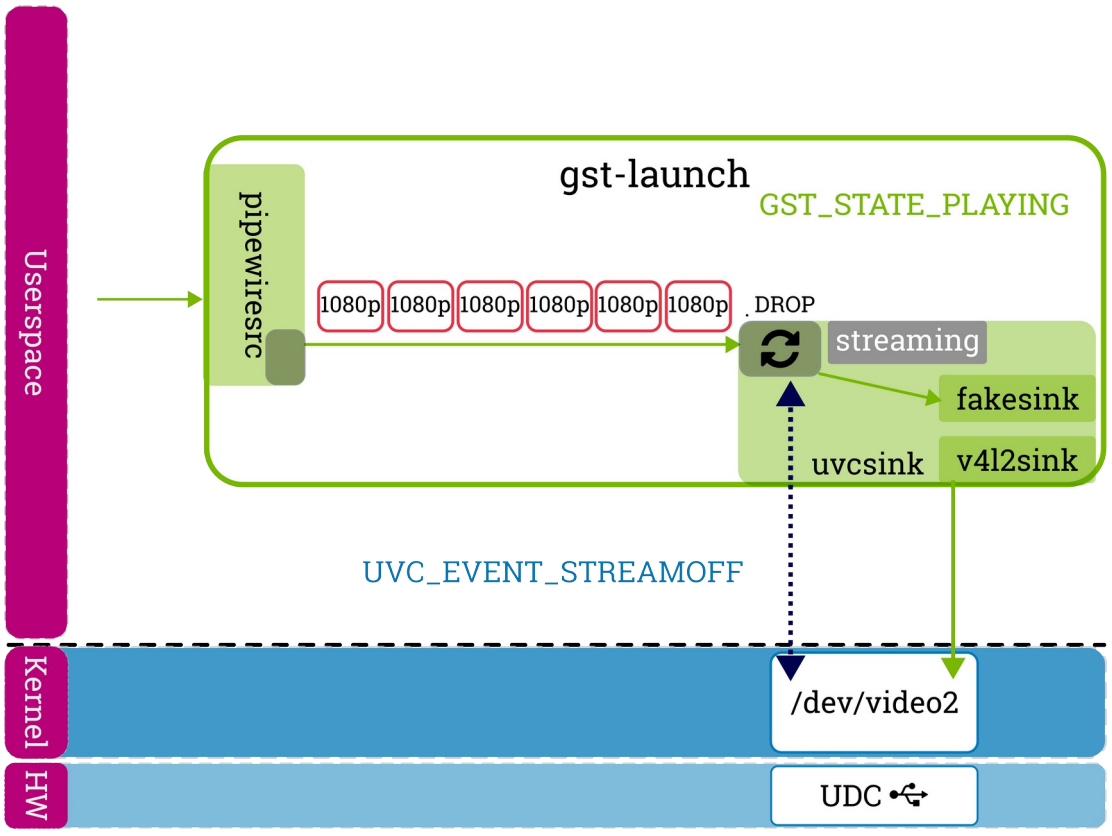
uvcsink – live mode



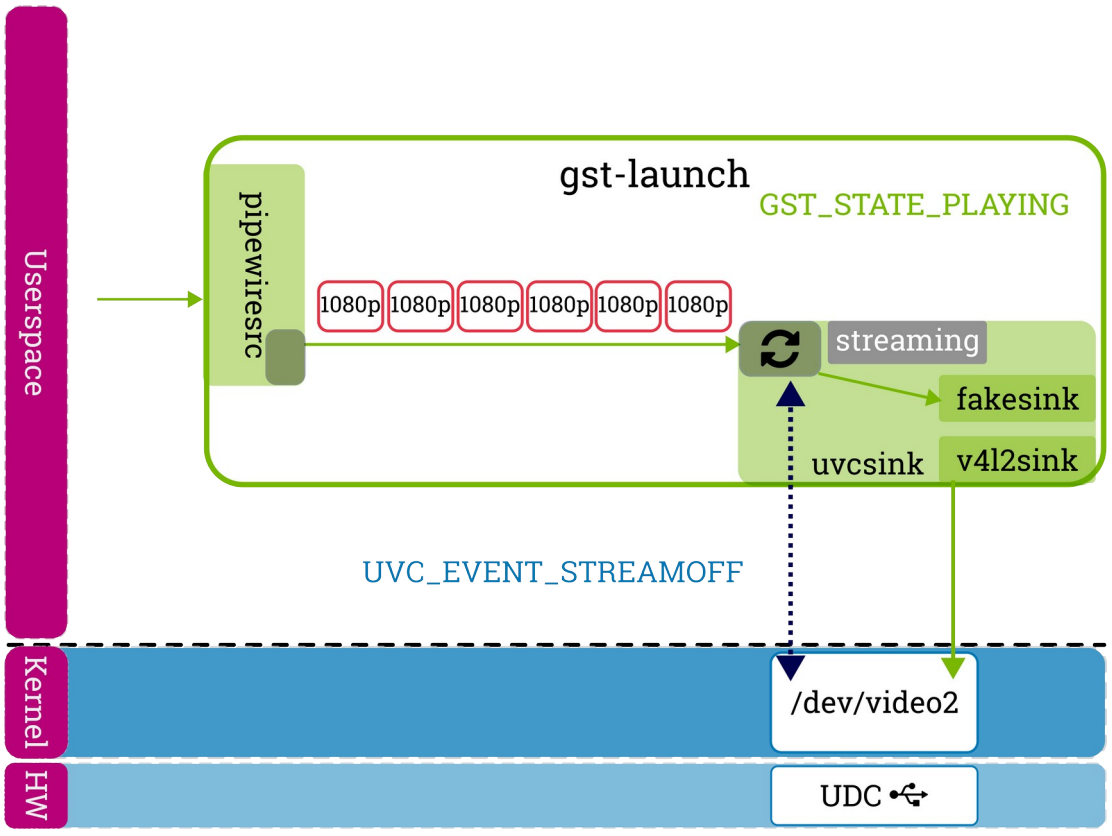
uvcsink – live mode



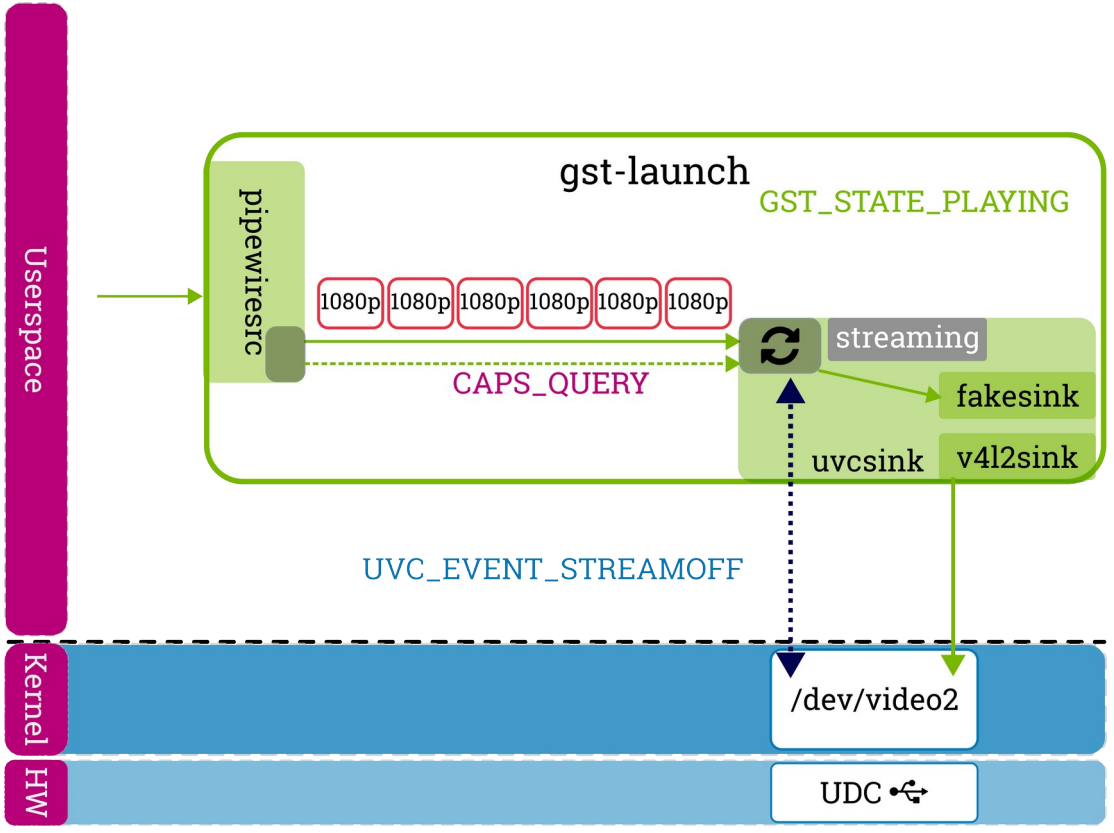
uvcsink – live mode



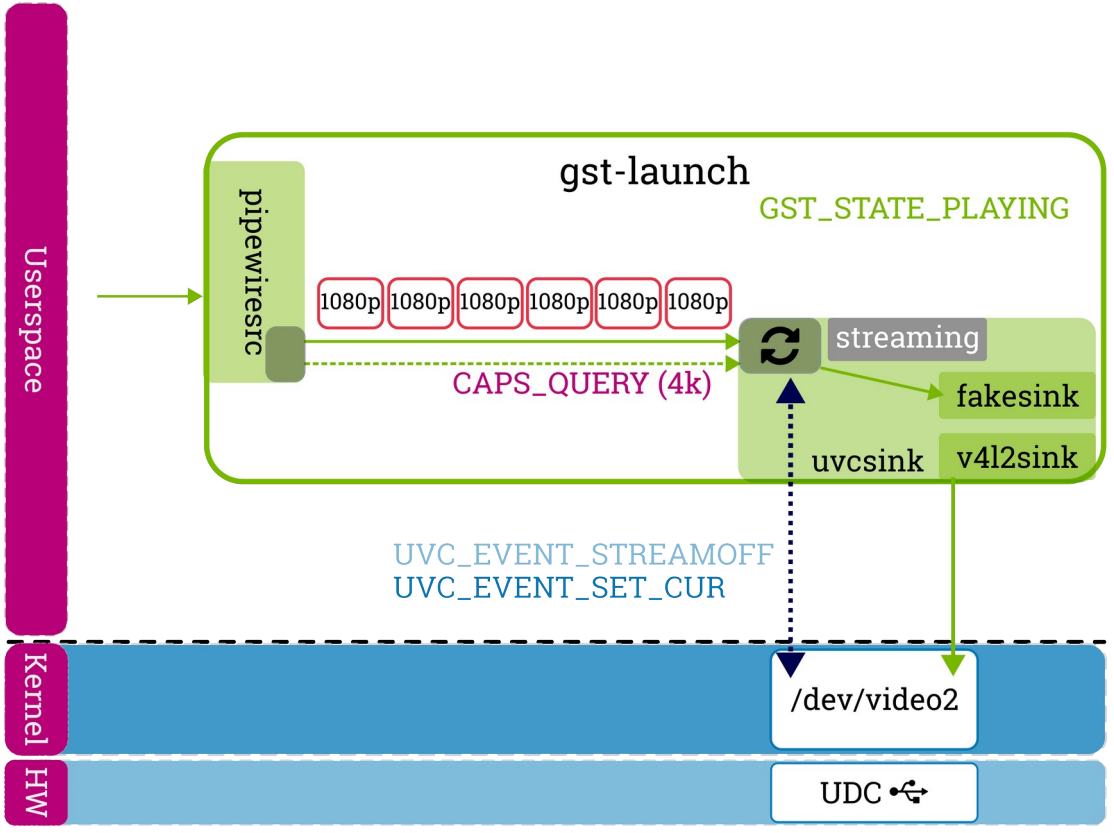
uvcsink – live mode



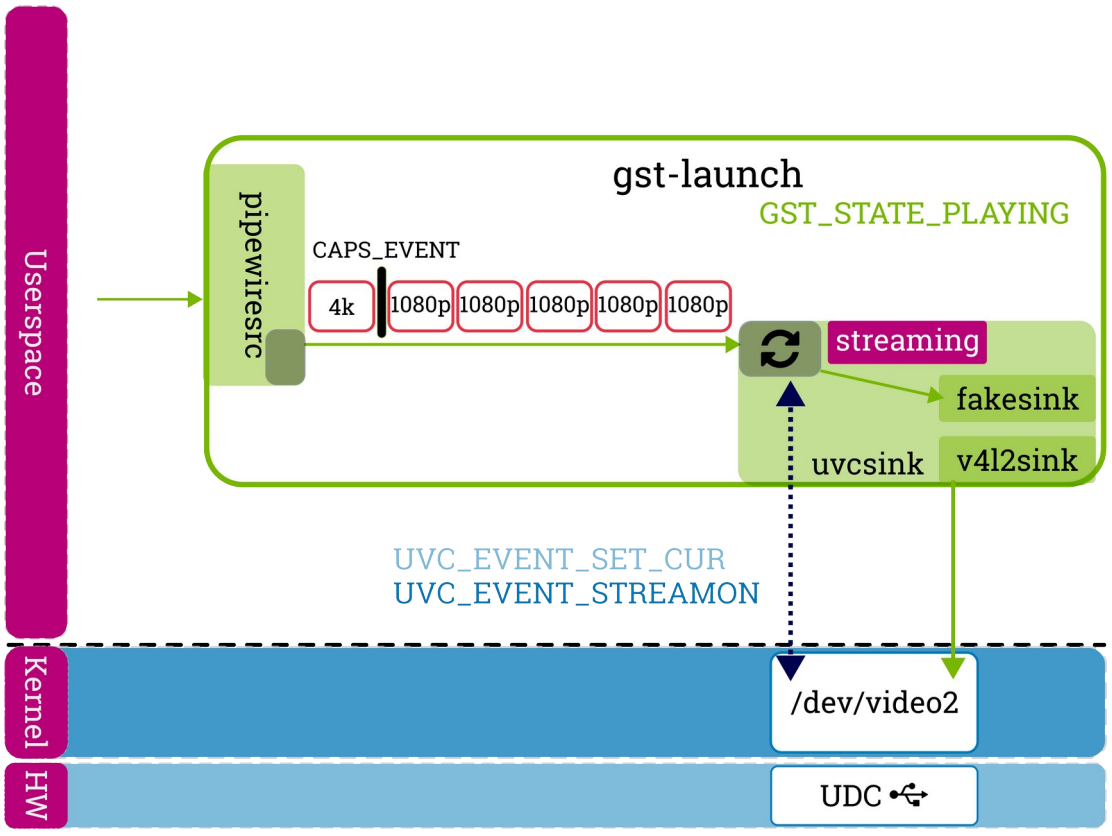
uvcsink – live mode



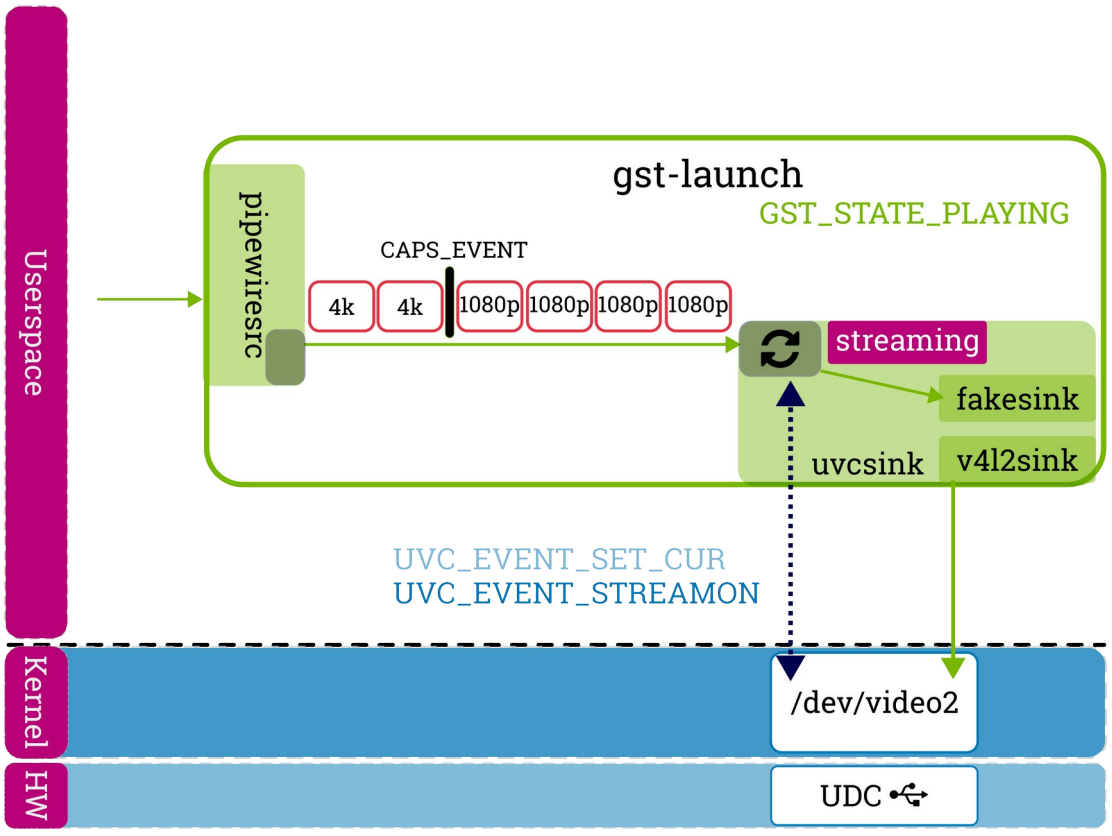
uvcsink – live mode



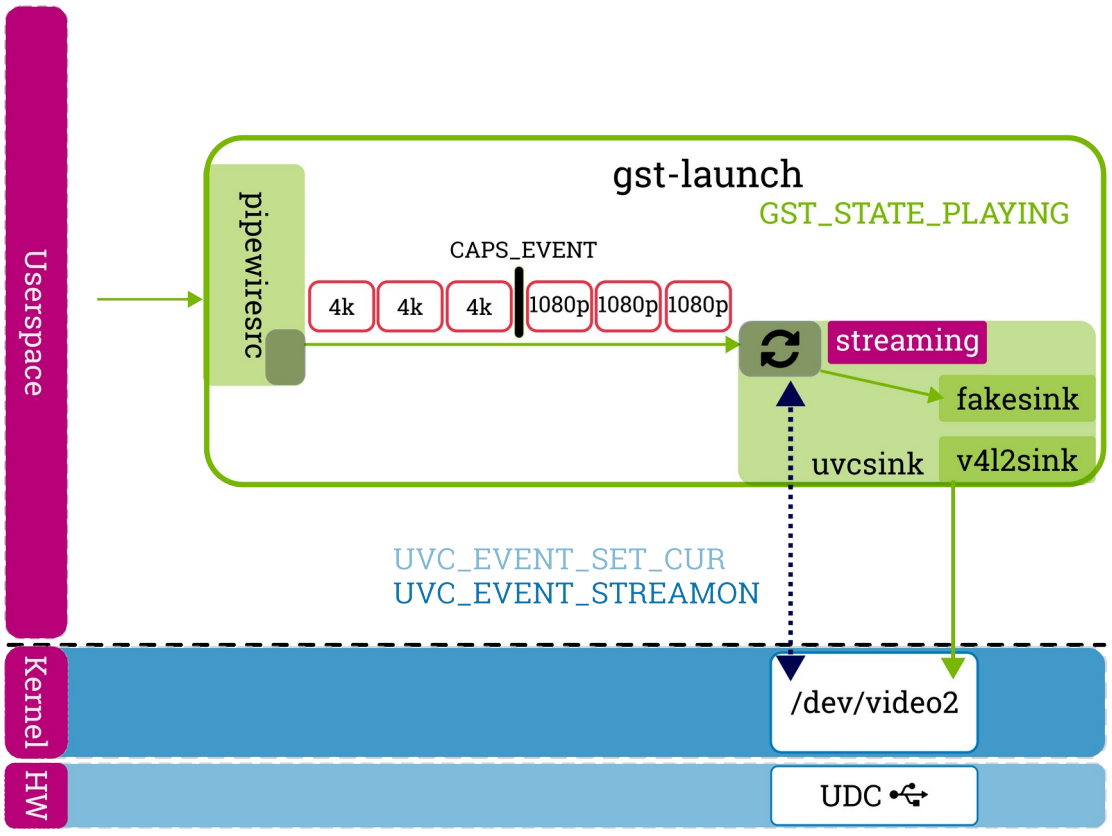
uvcsink – live mode



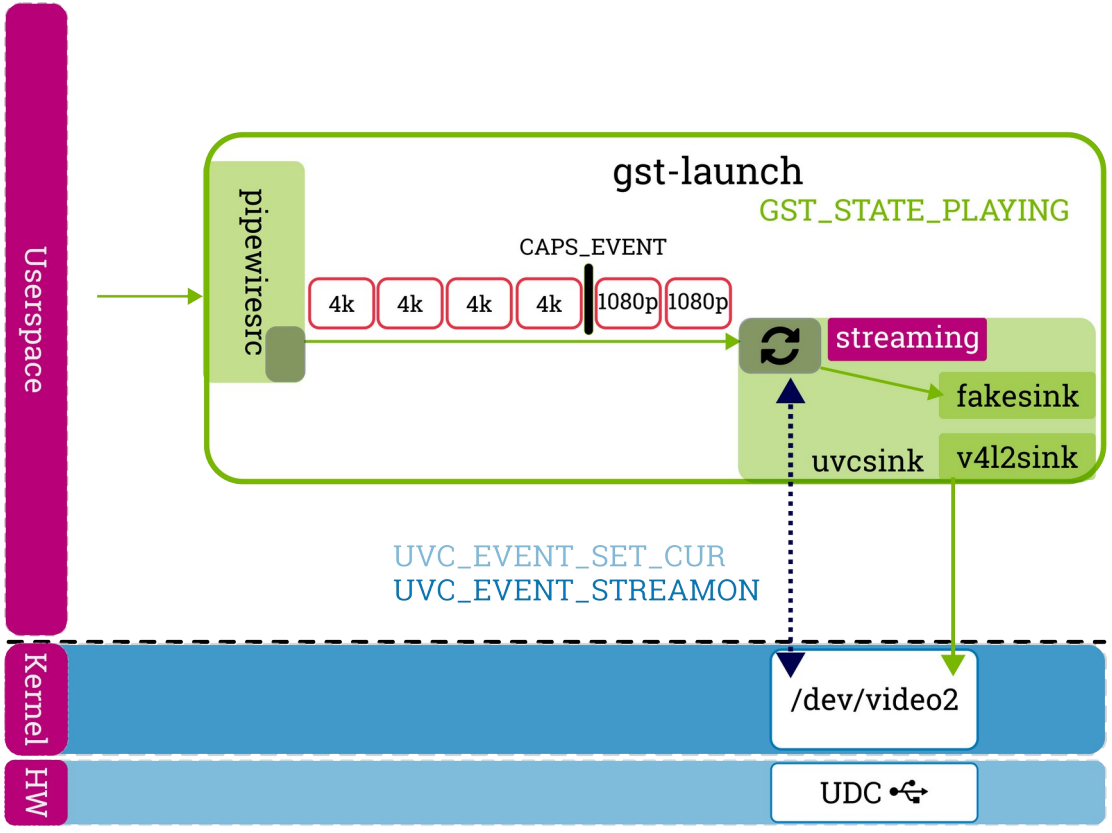
uvcsink – live mode



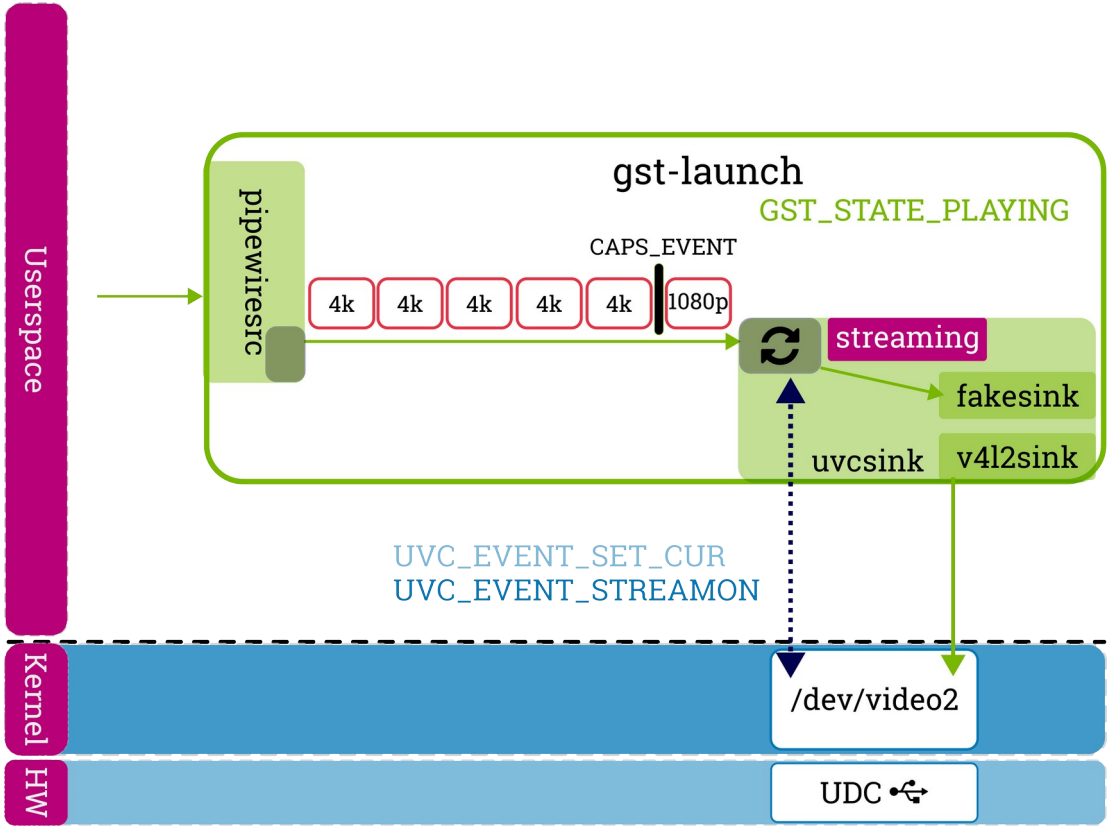
uvcsink – live mode



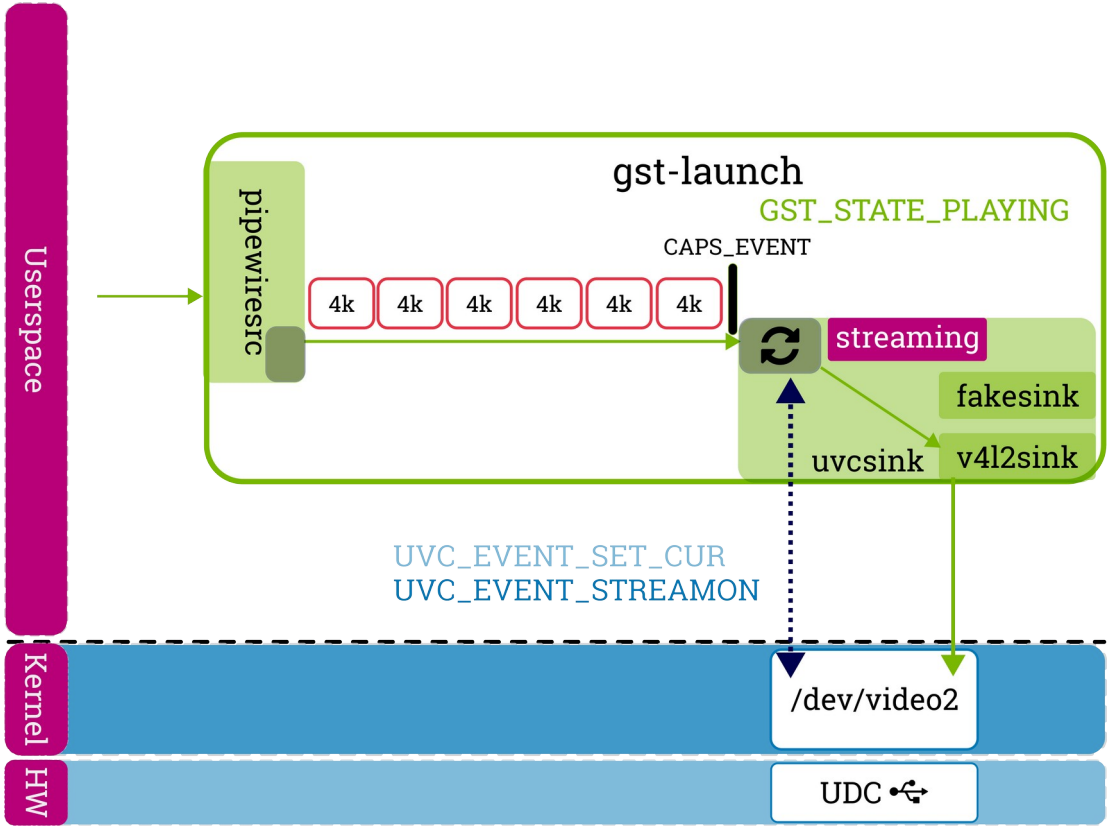
uvcsink – live mode



uvcsink – live mode



uvcsink – live mode



uvcsink – probes

- early idle probe is registered on GST_STATE_READY
  - create dropping buffer probe while relinking
- event probe on format change to catch CAPS_EVENT
  - dynamically created on CAPS_QUERY in the sink_query callback

uvcsink – event handling in padtask

- polling for host events
  - reused event parsing from uvc-gadget
- subscribe to UVC_EVENT on GST_STATE_READY
  - STREAMON, STREAMOFF, SETUP, DATA, DISCONNECT
- UVC_VS_COMMIT_CONTROL → fixate uvcsink caps
- UVC_EVENT_STREAMON/STREAMOFF
  - notify event for the streaming property change
  - toggle variables „streamon/streamoff“ for idle handling (live-mode)

Open Topics - Next steps

Open Topics

- Open MR for stability in live-mode
  - https://gitlab.freedesktop.org/gstreamer/gstreamer/-/merge_requests/5170

Next Steps

- gadget preparation
  - skip formats in schemefile
  - use libusbgx in uvcsink
  - parse the connected source caps
  - write format layout from uvcsink to configfs

```
formats :
{
 mjpeg :
 {
  frames = (
  {
   dwFrameInterval = 333333;
   wHeight = 1080;
   wWidth = 1920;
  }, {
   dwFrameInterval = 333333;
   wHeight = 2160;
   wWidth = 3840;
  } );
 };
};
```

- improve libuvcgadget api and use it

Thanks!

Thanks to Nicolas Dufresne

Compile it, Use it, Improve it!

Any Questions?